JLISNT Journal of Leading Innovations in Science and Technology
*Journal of Leading Innovations in Science and Technology, Vol.1, No.1, pp. 55-64, Jan. 2026*

# A MODULAR AND SCALABLE VLSI DESIGN FOR ROBUST OBJECT DETECTION AND TRACKING IN UNPREDICTABLE ENVIRONMENTS

Yasser Rahim[1], Dr. Vijeta Yadav [2], Dr. K.Venkata Murali Mohan [3]

[1]*Research Scholar, Madhyanchal Professional University (MPU), India.* yasserrahim1146@gmail.com,
[2]*Supervisor, Madhyanchal Professional University (MPU), India.* vijeta.cool17@gmail.com
[3]*Professor of ECE and Principal, Teegala Krishna Reddy, Engineering College, Medbowli, Meerpet(M), Ranga Reddy District, Hyderabad, Telangana-500097,* kvmmece@gmail.com

***Abstract*** – The development of digital image processing has been a significant contributor to the improvement of machine perception and the interpretation of complex scenes, particularly in environments that are dynamic and unpredictable. Applications such as autonomous navigation, surveillance, remote sensing, medical imaging, and industrial automation are just some of the many that require robust, real-time object detection and tracking systems. These kinds of systems are necessary in light of the circumstances. An adaptive Very Large Scale Integration (VLSI) architecture that is specifically tailored for robust object detection and tracking in dynamic environments is the goal of this study, which aims to propose such an architecture. To effectively manage variable resolution and rapid scene changes while maintaining computational efficiency, the proposed system incorporates a space-variant edge detection mechanism and an adaptive image scaling unit. This makes it possible for the system to effectively deal with both of these kinds of factors. Interpolation-induced blurring can be reduced, and edge precision can be improved through the use of a pre-filtering technique that makes use of spatial filters. Additionally, improved detection accuracy is achieved through the implementation of hardware-efficient bilinear interpolation in conjunction with adaptive pixel selection. The design makes use of hardware sharing techniques to reduce the number of gates and the area of silicon, thereby achieving an effective balance between performance and the utilization of resources. The synthesis of the architecture was accomplished through the utilization of the TSMC 0.13 µm CMOS process. It exhibits a processing rate of approximately 280 MHz, and it has a gate count of 6.67 K. This makes it suitable for real-time embedded vision systems. Cascading chips can further improve performance in complex scenarios thanks to the modular and scalable design, which also ensures adaptability to changing environmental conditions and application requirements.

***Keywords***: *Adaptive VLSI Architecture, Real-Time Object Detection, Object Tracking, Dynamic Environments, Space-Variant Edge Detection, Adaptive Image Scaling, Bilinear Interpolation, Spatial Filtering, Hardware Optimization, CMOS VLSI Design, Embedded Vision Systems, Low-Power Design, Gate Count Reduction, Image Processing Hardware, Scalable Architecture*

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com

*Accepted Jan.2026,* 

# I. INTRODUCTION

## 1.1 INTEGRATED CIRCUITS AND THEIR ROLE

Integrated Circuits (ICs), often termed microchips or monolithic integrated circuits, are compact assemblies of electronic circuits fabricated onto a single semiconductor substrate, predominantly silicon. The continuous miniaturization and integration of billions of transistors within a single chip have significantly transformed electronic system design, enabling enhanced operational performance, reduced power consumption, and minimized physical dimensions. These advancements are particularly critical in adaptive Very-Large-Scale Integration (VLSI) architectures, where high processing speeds and low latency are essential for real-time object detection and tracking systems.

In modern high-computation environments, ICs act as the hardware backbone for accelerated systems executing complex operations such as convolutional neural networks (CNNs), optical flow estimation, and feature-based tracking. Their scalability and reliability have led to the widespread adoption of reconfigurable and application-specific hardware platforms, including Field Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs), which are increasingly integrated into VLSI architectures tailored for intelligent vision systems.

Advancements in semiconductor fabrication, guided by Moore's Law, have allowed billions of transistors to be integrated within extremely compact chips. This increased computational density supports high-throughput parallel processing critical for real- time applications in autonomous vehicles, surveillance, and robotics. Moreover, IC-based VLSI systems enhance processing speed, energy efficiency, and adaptability to changing environmental conditions, making them ideal for vision-based applications requiring real-time responsiveness

| Name | Signification | Year | Transistors number | Logic gates number |
|---|---|---|---|---|
| VLSI | very large-scale integration | 1980 | 20,000 to 1,000,000 | 10,000 to 99,999 |
| ULSI | ultra-large-scale integration | 1984 | 1,000,000 and more | 100,000 and more |
| SSI | small-scale integration | 1964 | 1 to 10 | 1 to 12 |
| MSI | medium-scale integration | 1968 | 10 to 500 | 13 to 99 |
| LSI | large-scale integration | 1971 | 500 to 20,000 | 100 to 9,999 |

**Table 1.1**: Various IC's Signification and Configuration.

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com
*Accepted Jan.2026,* 

## II. BACKGROUND AND RELATED WORK

This chapter provides the foundational background necessary to understand the key concepts and technologies underpinning this research. The focus is on the domain of VLSI (Very Large Scale Integration) design, particularly within the context of adaptive architectures suited for real-time object detection and tracking in dynamic and unpredictable environments.

Electronic Design Automation (EDA) plays a pivotal role in enabling the efficient and accurate design of complex electronic systems. It encompasses a wide range of tools and methodologies aimed at automating both analog and digital system design. Within EDA, Computer-Aided Design (CAD) technologies are utilized to support and streamline electronic system design, allowing for faster iterations and higher precision.

A significant subdomain of EDA, and the core area of this research, is VLSI CAD. This field addresses the automation of the VLSI chip design process, including synthesis, placement, routing, verification, and optimization. With the increasing complexity of embedded and intelligent systems, VLSI CAD has evolved to support adaptive architectures that are capable of responding to real-time data and environmental changes.

This research situates itself at the intersection of VLSI CAD and real-time computer vision applications. The aim is to develop adaptive VLSI architectures that are not only efficient in terms of hardware resources but are also robust and scalable for performing object detection and tracking in dynamic and often unpredictable environments—such as those encountered in autonomous navigation, surveillance, and smart robotics.

### 2.1 HIGH-LEVEL SYNTHESIS

High-Level Synthesis (HLS), also known as C synthesis or behavioral synthesis, has evolved as a key methodology for translating algorithmic models into optimized digital hardware architectures. In the context of adaptive VLSI systems for real- time object detection and tracking, HLS significantly enhances the efficiency of hardware prototyping and implementation of complex vision and signal processing algorithms.

HLS enables system designers to describe functionality using high-level programming languages like C, C++, System C, and MATLAB, eliminating the complexities of manual low-level hardware description [46]. This abstraction focuses design efforts on system accuracy and adaptability, crucial for applications involving dynamic environmental conditions such as motion blur or varying illumination. Subsequently, the high-level descriptions are converted into Register Transfer Level (RTL) code, and ultimately into gate-level net lists through standard synthesis workflows.

By supporting design exploration and rapid iterations, HLS allows developers to optimize

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com
*Accepted Jan.2026,* 

architectures concerning power consumption, silicon area, and computational efficiency—factors vital for embedded vision applications. Fixed-point implementation of algorithms like edge detection and region segmentation further improves computational efficiency without sacrificing output accuracy.

The automated HLS flow encompasses algorithmic analysis, control and dataflow graph generation, scheduling, resource sharing, and synthesizable RTL code generation with cycle-accurate behavior, thereby enabling efficient hardware realization suitable for real-time adaptive object tracking.

With the increasing transistor density of modern semiconductor technologies, HLS offers scalable, modular, and reusable architectures necessary for implementing functional units such as adaptive thresholding and motion estimation with minimal design overhead.

## 2.2 SCHEDULING PROBLEM

In the continuously evolving landscape of adaptive Very Large-Scale Integration (VLSI) architectures, especially in applications like robust object detection and real- time tracking, scheduling plays a pivotal role in determining the sequence of operations while adhering to stringent design constraints.

At its core, the scheduling problem involves systematically allocating operations derived from Hardware Description Language (HDL) codes into well-defined control steps, thereby ensuring compliance with factors such as data dependencies, hardware resource availability, and specific system limitations. Each control step typically aligns with a single clock cycle, within which operations are executed either concurrently or sequentially, subject to hardware and dependency constraints.

Recent advancements highlight that scheduling holds critical importance within High- Level Synthesis (HLS) processes, where it not only determines the precise timing of each operation's execution but also significantly influences major design trade-offs, including delay minimization, area reduction, power efficiency, and throughput optimization.

Let represent the set of operations derived from the HDL model. If operation relies on the result of a prior operation of, then it must await the completion of enforcement of precedence constraints that preserves logical execution sequences [52]. These inter-operation dependencies are typically represented using Directed Acyclic Graphs (DAGs), which serve as foundational models for schedulers to guarantee computational correctness.

## 2.3 OBJECT TRACKING USING BLOCK MATCHING ALGORITHM

Object detection and tracking are fundamental components of modern computer vision systems, particularly in applications that operate within dynamic and real-time environments. These tasks are critical for a wide range of applications including autonomous vehicles, intelligent surveillance, traffic

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com
*Accepted Jan.2026,* 

monitoring, robotic navigation, military reconnaissance, and human–machine interaction systems. Despite significant advancements in visual processing algorithms, achieving robust and reliable tracking remains challenging due to factors such as illumination variation, background clutter, rapid object motion, scale changes, and partial or complete occlusion of targets.

In recent years, the growing adoption of embedded and edge-based vision systems has shifted research focus toward lightweight and energy-efficient hardware architectures capable of supporting real-time processing under strict computational and power constraints . Traditional software-based tracking implementations executed on general-purpose processors often fail to satisfy real-time requirements, especially when processing high-resolution video streams. Consequently, adaptive VLSI-based hardware acceleration has emerged as a viable solution for meeting performance, latency, and power efficiency demands.

### 2.4 BLOCK MATCHING APPROACH

Block Matching is a classical and widely adopted technique for motion estimation, forming the foundation of many video processing, compression, and real-time object tracking systems.

The primary objective of this approach is to estimate motion vectors that describe the displacement of image regions between consecutive video frames. Due to its simplicity, robustness, and suitability for hardware realization, block matching continues to be extensively used in surveillance, embedded vision, and adaptive tracking applications.

In a typical Block Matching Algorithm (BMA), each video frame is partitioned into smaller, fixed-size, non-overlapping square regions known as macro blocks. For every macro block in the current frame, a corresponding best-matching block is searched within a defined search window in the reference (previous) frame.

The similarity between blocks is evaluated using cost functions such as the Sum of Absolute Differences (SAD) or Mean Squared Error (MSE). The block position that minimizes the selected cost metric determines the motion vector, representing the spatial displacement of the macro block between frames.

### 2.5 EUCLIDEAN DISTANCE

In object detection and tracking within dynamic visual environments, accurate association of features across successive frames relies heavily on reliable distance computation techniques. Among the various similarity measures employed, Euclidean distance remains one of the most fundamental and widely used metrics for quantifying spatial similarity or dissimilarity between feature points. Euclidean distance measures the direct ―straight-line‖ separation between two points in Euclidean space,

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com
*Accepted Jan.2026,* 

representing the shortest possible path connecting them. Owing to its mathematical simplicity and low computational overhead, it is particularly well suited for real-time applications implemented on adaptive VLSI architectures, where processing efficiency and hardware resource constraints are critical considerations.

Consider two points X and Y in a two-dimensional Euclidean space, where

X = (x1, x2) and Y = (y1, y2).

The line segment joining these points forms the hypotenuse of a right-angled triangle. Applying the Pythagorean Theorem, the Euclidean distance between the two points is computed as:

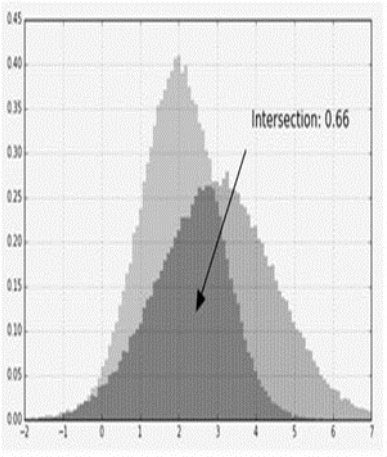
$d(X, Y) = \sqrt{(x1 - y1)^2 + (x2 - y2)^2}$….1

## 2.6 HISTOGRAM INTERSECTION

In object detection and tracking applications operating within dynamic visual environments, histogram-based similarity measures have emerged as computationally efficient and hardware-friendly techniques for evaluating frame-to-frame consistency. A histogram provides a compact statistical representation of pixel intensity or color distributions within a digital image by capturing the frequency of occurrence of grayscale or chromatic values. Such statistical abstraction is particularly valuable for object representation and matching in real-time vision systems implemented using adaptive VLSI architectures, where memory usage and processing efficiency are critical constraints.

Among various histogram-based similarity measures, histogram intersection is one of the most widely adopted techniques for quantifying the degree of similarity between two images or regions. This method evaluates the amount of overlap between two discrete histogram distributions, with the intersection value ranging from 0, indicating no similarity, to 1, indicating a perfect match . In object tracking systems, a higher histogram intersection value reflects stronger resemblance between the current

Figure 2 illustrates the concept of histogram intersection, where overlapping regions between two histograms represent shared intensity distributions contributing to the overall similarity score.




Corresponding Author: Yasser Rahim
yasserrahim1146@gmail.com
Accepted Jan.2026, 

## III. PROPOSED ARCHITECTURE FOR BLOCK MATCHING

In the evolving domain of dynamic scene analysis, block matching continues to serve as a foundational technique for real-time object tracking. The architecture proposed in this study is designed as a self-contained and autonomous framework, aligning with the core objectives of adaptive VLSI architectures that demand flexibility, robustness, and high tracking accuracy in complex and dynamic environments. Object tracking within the proposed system is achieved through temporal association, wherein a bounding rectangle is initially established around the target object in a reference frame. Subsequently, tracking is performed automatically using low-level processing techniques such as motion detection and block-wise pixel intensity matching across successive frames, thereby eliminating the need for manual intervention. The proposed architecture processes continuous video streams, typically originating from surveillance and monitoring scenarios that require dynamic adaptation and precise object localization. Prior to block matching, a motion detection stage is employed to identify the Region of Interest (ROI). This stage utilizes frame differencing and background subtraction techniques to isolate moving objects from static background regions, ensuring that subsequent computations are confined to relevant areas of the frame. Once the ROI is detected, it is partitioned into uniformly sized, non-overlapping blocks, which enhances tracking precision while reducing computational overhead.

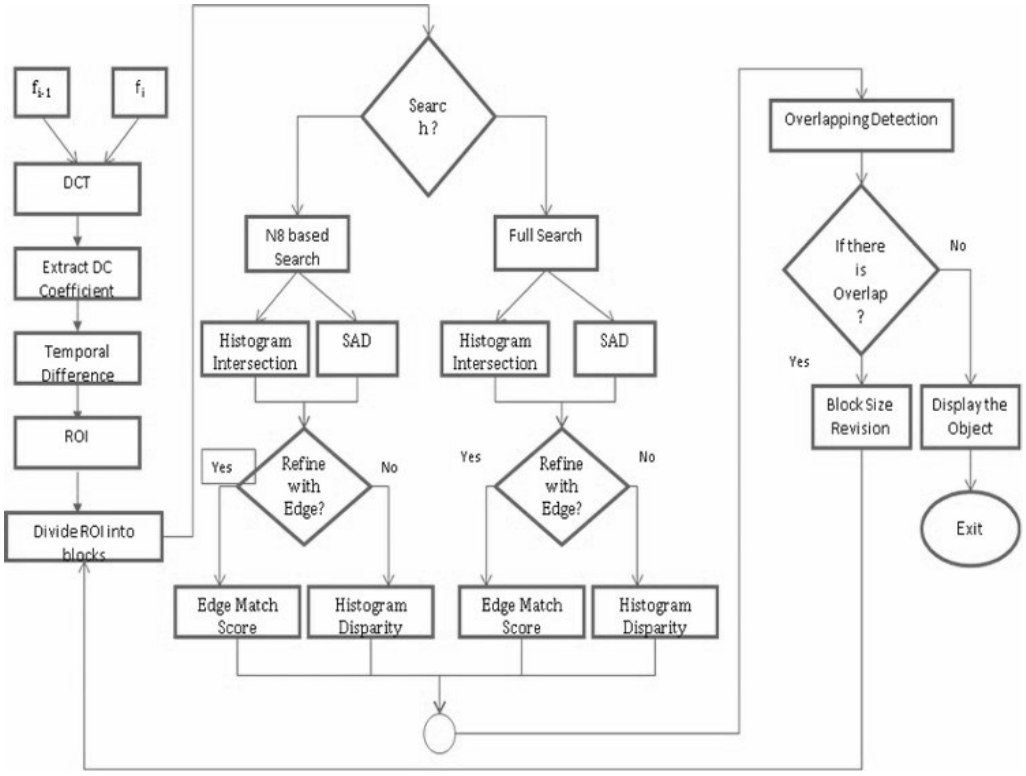



Figure 1: Block matching approach framework for object tracking.

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com

*Accepted Jan.2026,* 

## IV. RESULTS&DISCUSSION

To evaluate the effectiveness and robustness of the proposed adaptive VLSI-based object tracking framework, extensive experimental validation was carried out using IDL 6.3 on a Windows platform. The primary objective of this evaluation was to verify the reliability of the proposed block-matching-based tracking algorithm under both controlled and real-world dynamic conditions, thereby assessing its suitability for deployment in practical embedded vision systems.

The experimental analysis began with an ideal or controlled test scenario, illustrated in Figure 4.6, where the target object remained continuously visible without occlusion, background interference, or illumination variation. This baseline experiment established a reference performance benchmark and confirmed the correctness of the proposed tracking pipeline under noise-free conditions.

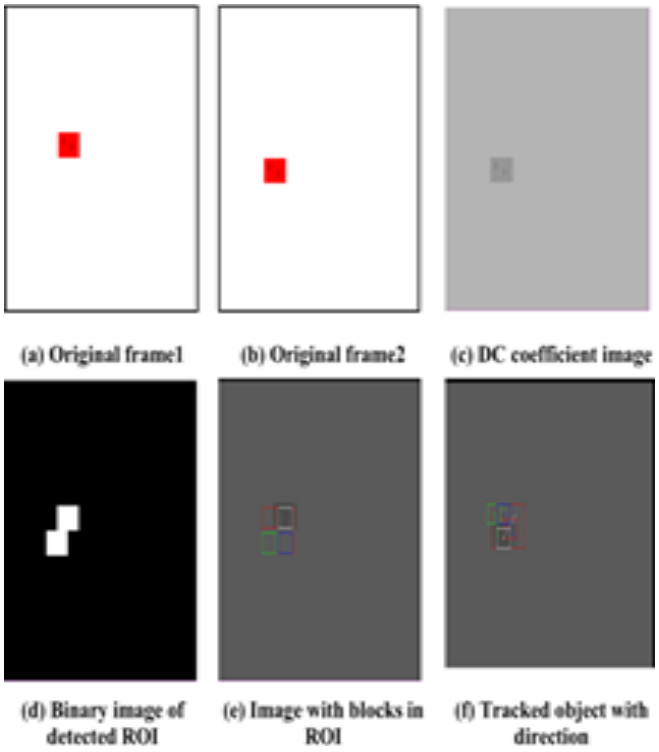



**Figure 2:** Tracking the Perfect Example.

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com
*Accepted Jan.2026,* 

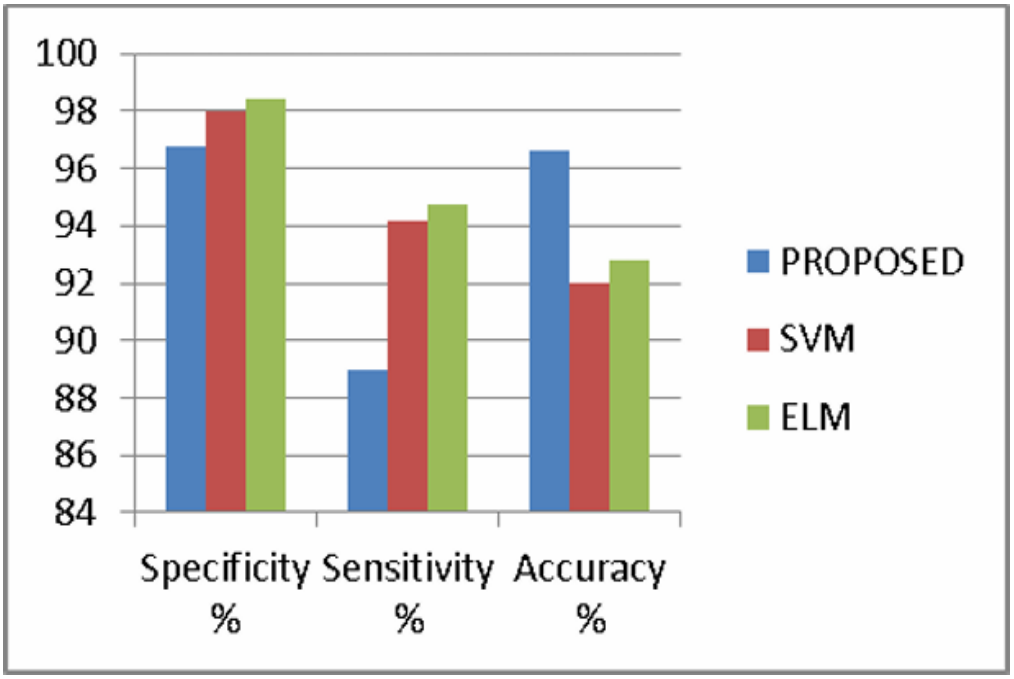



**Table .1:** Alternative approaches compared

## V. CONCLUSION

This research proposed an adaptive, low-complexity VLSI architecture tailored for efficient object detection and tracking in dynamic and high-resolution environments. The core contribution lies in the design of an edge-preserving, block-partitioned detection algorithm (ABPD) that subdivides image blocks into four sub-blocks, effectively minimizing edge distortion and enhancing detection robustness. Simulation outcomes demonstrate that the proposed method significantly outperforms conventional low-complexity designs in terms of adaptability, object similarity, and processing efficiency. Furthermore, the hardware implementation of the ABPD algorithm enables high-speed, cost-effective performance suited for real-time 4K2K video applications. The developed architecture showcases superior detection accuracy, reduced computational overhead, and enhanced scalability, positioning it as a viable solution for next-generation embedded vision systems and edge-based AI applications.

**Funding:** This research received no external funding

**Data Availability Statement:** No new data were generated during the study. All the data are contained within the manuscript.

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com

*Accepted Jan.2026,* 

**Acknowledgments:** We acknowledge the use of Grammarly and Quill Bot, in correcting the English and Grammatical errors in the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Ethics Declaration:** This manuscript is a review article and does not involve any studies with human participants or animals performed by the authors. Therefore, ethical approval and informed consent were not required. The authors declare no conflict of interest, and all referenced works have been properly cited.

## REFERENCES

[1]. S. Mohammed Sali, M. Meribout, A. Abdul Majeed, "Real Time FPGA Based CNNs for Detection, Classification, and Tracking in Autonomous Systems: State of the Art Designs and Optimizations," arXiv:2509.04153, 2025.

[2]. .Montgomerie-Corcoran, P. Toupas, Z. Yu, C-S. Bouganis, "SATAY: A Streaming Architecture Toolflow for Accelerating YOLO Models on FPGA Devices," arXiv:2309.01587, 2023.

[3]. K. Kumar, D. Nandan, R. K. Mishra, "Compact Hardware of Running Gaussian Average Algorithm for Moving Object Detection Realized on FPGA and ASIC," Research in India, 33(4), pp. 681–689, 2019.

[4]. W. Liu, S. Zhang, and Z. Huang, "Deep Learnig for Generic Object Detection: A Survey," International Journal of Computer Vision, vol. 128, no. 2, pp. 261–318, 2020.

[5]. Z. Wang, H. Li, and J. Zhao, "Hardware-Efficient Object Detection Accelerator for Edge AI Applications," IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 7, pp. 4281–4294, 2022.

[6]. S. Devi Poonguzhali, P. Nancy, M. Swathi, M. Vidhya, "VLSI Implementation of Object-Detection Design Using Adaptive Block Partition Decision Algorithm," IJERT 2017.

[7]. C. Li, R. Xu, Y. Lv, Y. Zhao, W. Jing, "Edge Real-Time Object Detection and DPU- Based Hardware Implementation for Optical Remote Sensing Images," Remote Sensing, vol. 15, no. 16, 2023.

[8]. K. Kumar, R. K. Mishra, D. Nandan, "Efficient Hardware of RGB to Gray Conversion Realized on FPGA and ASIC," Procedia Computer Science, 171, pp. 2008-2015, 2020.

[9]. Osman, I., "BgSub: A Background Subtraction Model for Effective Moving Object Detection,"

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com

*Accepted Jan.2026,* 

ACCV Workshop, 2024.

[10]. Al-Yoonus, M. A., "FPGA-Based Implementation of a Basic Background Subtraction Algorithm," IJEEE, 2025. This paper implements a block-region background subtraction (BGS) algorithm on FPGA hardware as preprocessing for object detection.

[11]. Oshima, Y., Yamaguchi, Y., Tsugami, R., Fujiwara, T., Fukui, T., Narikawa, S., "FPGA-based Improved Background Subtraction for Ultra-Low Latency," IEEE Access, 2025. They present a background removal / subtraction implementation on FPGA targeting very low latency, relevant for streaming and critical applications.

[12]. "Motion Object Tracking System Based on FPGA" (authors not listed), published recently. This reports block-based motion estimation using SAD between reference and current blocks for object tracking.

*Corresponding Author:* Yasser Rahim
yasserrahim1146@gmail.com

*Accepted Jan.2026, © 2026 JLISNT*